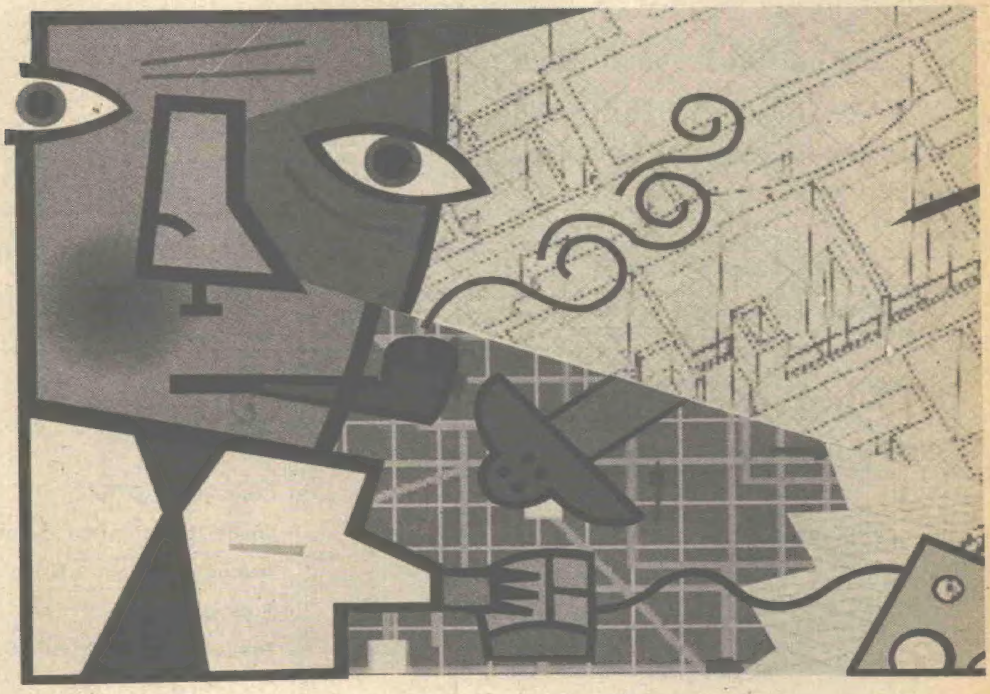
วิทยา สงวนวรรณ

เทคนิค AutoCAD : การจัดการฐานข้อมูล ด้วย ASE ตอนที่ 2



อธิบายการนำเทคนิค ASE มาใช้ในการจัดการฐานข้อมูลบนออโต้แคด เพื่อการใช้งานอย่างเต็มประสิทธิภาพ

ในฉบับที่แล้ว ผมได้พูดถึงเรื่อง *การจัดการฐานข้อมูล (Database) บน AutoCAD ด้วย ASE* แต่ยังค้างติงตรงที่การเข้าไปวิวดูฐานข้อมูลที่ลิงค์กับ Entity บนจอภาพ ฉบับนี้ก็คงเป็นตอนที่ 2 ต่อเนื่องจากฉบับที่ 100 สำหรับท่านผู้อ่านที่เริ่มอ่านคอลัมน์นี้ก็ควรกลับไปดูเรื่องราวในฉบับที่ 100 ประกอบด้วย จะทำให้ท่านเข้าใจการทำงานของ ASE ได้ดียิ่งขึ้นครับ

คำถาม

มีผู้อ่านหลายท่านได้ติดต่อโดยตรงกับผม และสงสัยเรื่องราวที่เกี่ยวกับดาต้าเบสที่ยังไม่กระจ่างชัดนัก พอคุยกันไปสักพักก็ถึงรู้ว่า ท่านผู้อ่านท่านนั้นนำไปปนกับดาต้าเบสใน Drawing ของออโต้แคด ก็ขอบอกได้เลยว่าดาต้าเบสที่ผมกล่าวถึงทั้งหมดในหัวข้อ ASE นี้จะเป็นดาต้าเบสที่อยู่ภายนอกออโต้แคด หรือที่เรียกกันว่า *"External Database"* เท่านั้น ซึ่งก็หมายถึงดาต้าเบสที่สร้างขึ้นมาจากโปรแกรมอื่น (ประเภท DBMS) ว่ากันง่าย ๆ ก็คือ ไม่ใช้ file Drawing ของเราที่มีนามสกุล .DWG เลยครับ และ External Database นี้จะจำเพาะของใครของมันตามแต่ละโปรแกรมจะสร้างกันขึ้นมา ส่วนดาต้าเบสของ Drawing ก็เป็นอีกเรื่องหนึ่งที่จะพูดถึงเมื่อมีตอนต่อ ๆ ไปในหัวข้อ Auto LISP และขอย้ำด้วยว่า ส่วนที่ท่านนำไปลิงค์กับ Entity นั้น จะต้องเป็น External Databaes นะครับ

เข้าใจให้กระจ่าง

ไหน ๆ ก็วกออกไปยังเรื่องราวของดาต้าเบสใน Drawing แล้วก็ขอกล่าวถึงเรื่องนี้อีกสักเล็กน้อย จะได้สร้างความกระจ่างให้ท่าน แล้วมาลองเปรียบเทียบกับดาต้าเบสของ dBASE 3 จะได้รู้ว่าลักษณะหน้าตาของการเก็บข้อมูลแตกต่างกันอย่างไรบ้าง ก่อนอื่นต้องเข้าใจว่าข้อมูล Drawing ของท่านนั้นจะถูกจัดเก็บอยู่ในรูปของเวกเตอร์ แต่ก็ใช่ว่าเป็นเวกเตอร์เลยทีเดียว ต้องมีการคำนวณแปลงข้อมูลที่เก็บไว้ให้เป็นเวกเตอร์อีกทีหนึ่งก่อน ถึงจะแสดงบน

จอภาพได้

เราลองมาสร้างตัวอย่างง่าย ๆ ขึ้นบนออโต้แคดโดยให้สร้าง Entity บนจอภาพสัก 2 Entity ที่เป็นเท็กซ์และวงกลม จากคำสั่ง DTEXT และ CIRCLE ดังในรูปที่ 1 โดยให้เท็กซ์อยู่ใน layer *"object1"* และวงกลมสร้างอยู่ใน layer *"object2"* ดังแสดงในรูปที่ 2

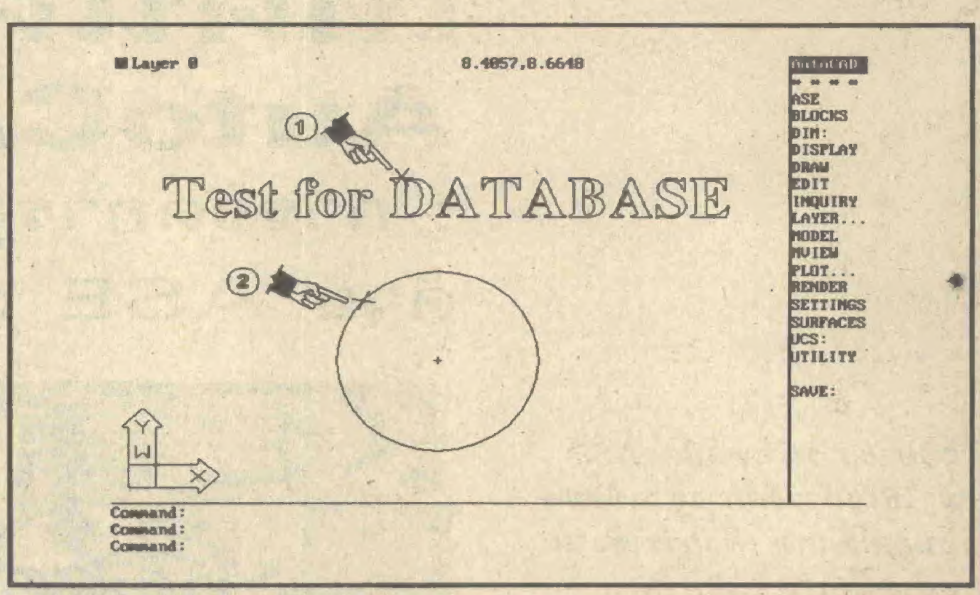
เมื่อเป็นที่เรียบร้อยแล้วก็ให้พิมพ์คำสั่ง AutoLISP เพื่อให้แสดงดาต้าเบสของทั้งสอง Entity นี้ทาง Command line

```
Command: (entget (ssname
(ssget 0))          ENTER
```

โปรแกรมจะให้ทำการ *"Select object:"* ก็ให้ Pick ไปยังตำแหน่งที่ 1 ก็จะปรากฏข้อมูลดาต้าเบสของ Entity ที่เป็นเท็กซ์แสดงออกมาทาง Command line

และในทำนองเดียวกันก็ให้พิมพ์คำสั่งเดิมอีกครั้งแต่คราวนี้ไป Pick ในตำแหน่งที่ 2 แทน เพื่อเปรียบเทียบดาต้าเบสของทั้งสอง Entity และจะขออธิบายข้อมูลในแต่ละวงเล็บดังแสดงในรูปที่ 3

ซึ่งทั้งหมดนี้จะเป็นลักษณะการเก็บข้อมูลดาต้าเบสของ Drawing ที่ต้องใช้ฟังก์ชันใน AutoLISP ดึงออกมา หรือที่เรียกว่า *ข้อมูลแบบลิสต์* แต่ที่น่าสนใจจะอยู่ตรงที่ว่าท่านสามารถเข้าไปอีดิตปรับเปลี่ยนข้อมูลในแต่ละวงเล็บได้อย่างง่าย ๆ เพียงใช้ฟังก์ชันใน AutoLISP ยกตัวอย่าง ถ้าท่านสร้างเท็กซ์และวงกลมที่ความสูง 1 หน่วย อยู่ใน layer "Text" สีแดง ก็จะให้มีการเปลี่ยนความสูงไปเป็น 1.5 หน่วย



รูปที่ 1 ตัวอย่าง Entity ที่สร้างขึ้นบนจอภาพ

```
(1) Command : LAYER                ; ใช้ตัวเลือก M สร้าง layer
                                     "object1"
(2) Command : DTEXT                ; เขียนตัวอักษรลงใน Drawing
    Justify/Style/<Start point>    ; Pick ที่ (1.2, 8)
    Height <0.00> : 1.53           ; ป้อนความสูง 1.53 หน่วย
    Rotation angle <0> :           ; Enter
    Text : Test for DATABASE       : ป้อนข้อความ
(3) Command : LAYER                ; ใช้ตัวเลือก M สร้าง layer
                                     "object2"
(4) Command : CIRCLE               ; ต้องการสร้างวงกลม
    3P/2P/TTR/<Center point>:      ; Pick ที่ (8.2, 4.1)
    Diameter/<Radius>: 2.5         ; ป้อนค่ารัศมี 2.5 หน่วย
```

รูปที่ 2 แสดงตำแหน่งของเท็กซ์และวงกลมว่าจะอยู่ ณ ที่ใด

พร้อมกับเปลี่ยน layer ไปเป็น layer อื่นสีเขียว หรือวงกลมจากรัศมีเดิม 1 หน่วย เปลี่ยนไปเป็นรัศมี 0.5 หน่วย พร้อมย้ายตำแหน่งก็ยังได้

ทั้งหมดนี้ก็พอจะช่วยท่านมองภาพดาต้าเบสของออโต้แคดกระจ่างชัดขึ้นนะครับ ถ้าหากท่านต้องการเข้าใจลึกซึ้งยิ่งกว่านี้ ก็คงต้องไปศึกษาเพิ่มเติมได้จากหนังสือคู่มือ AutoLISP แต่ต้องฉบับภาษาอังกฤษ หรือไม่ก็หนังสือ Database AutoCAD ก็ได้ แต่ถ้าอดใจรออีกสักนิด ผมจะนำเรื่องราวของ AutoLISP แบบมัน ๆ มาเสนอในโอกาสต่อไปครับ

```
( (-1. <Entity name :60000022>)   ; ชื่อของ Entity ที่ท่านเลือก
  ( 0."TEXT")                     ; ถูกสร้างขึ้นจากคำสั่ง Text
  ( 8."OBJECT1")                  ; อยู่ใน layer ที่ชื่อ "object1"
  (10 1.2 8.0 0.0)                ; ตำแหน่งวางข้อความที่ xyz
  (40. 1.56)                      ; ความสูงของตัวอักษร
  (1 . "Text for DATABASE")       ; ข้อความในเท็กซ์
  (50. 0.0)                       ; มุมองศาที่หมุนข้อความ
  (41. 1.0)                       ; ค่าความสัมพันธ์ Scale factor
  (51. 0.0)                       ; มุมของการเอียงตัวอักษร
  (7."ROMANTICBOLD")              ; รูปแบบตัวอักษร
  ...)
  ; ดาต้าเบสที่เป็นวงกลม
( (-1.<Entity name : 60000044>)   ; ชื่อของ Entity ที่ท่านเลือก
  ( 0."CIRCLE")                   ; ถูกสร้างขึ้นจากคำสั่ง CIRCLE
  ( 8."OBJECT2")                  ; อยู่ใน layer ที่ชื่อ "object2"
  (10 8.2 4.1)                    ; ตำแหน่งเซนเตอร์ของวงกลม
  (40. 2.50)                      ; ระยะรัศมีของวงกลม
  (210 0.0 0.0 1.0))
```

รูปที่ 3 อธิบายข้อมูลในแต่ละวงเล็บของดาต้าเบสที่เป็นเท็กซ์และดาต้าเบสที่เป็นวงกลม

การ Link Entity กับดาต้าเบส

คราวก่อนเป็นการกล่าวถึงเรื่องของดาต้าเบสเพียงเล็กน้อย ในตอนนี้ถึงคราวที่จะต้องกล่าวถึงกันอย่างจริงจังแล้ว สมมติให้ว่าขณะนี้ท่านสามารถสร้างดาต้าเบสได้ด้วย dBASE 3 ด้วยตัวท่านเองเป็นที่เรียบร้อยแล้ว และในตัวโปรแกรม AutoCAD Release 12 นี้ก็ได้สร้าง External Database (ต่อไปจะเรียกสั้น ๆ ว่าดาต้าเบส) มาให้ท่านอยู่แล้วทั้งหมด 3 ไฟล์ด้วยกัน โดยสร้างขึ้นจาก DBMS ที่ชื่อว่า dBASE 3 ทำให้สะดวกและรวดเร็วขึ้น (จะขอข้ามขั้นตอนที่ 2 ในหัวข้อ "การสร้างฐานข้อมูล" ไปเลย) คือ *INVENTRY.DBF, COMPUTER.DBF* และ *EMPLOYEE.DBF* ซึ่งเป็น Table ที่เป็นส่วนประกอบย่อยของดาต้าเบสที่ชื่อว่า *"ASETUT"*

การลิงค์เรคอร์ดใน Table เข้ากับ Entity เป็นวิธีการที่สำคัญของการทำ ASE มากที่สุด ถึงแม้ว่าขั้นตอนวิธีการทุกอย่างถูกต้อง แต่มีการลิงค์อย่างผิด ๆ ข้อมูลที่ค้นหาก็ผิดตามกันไปทั้งหมด และในขั้นตอนนี้จะต้องกระทำอย่างระมัดระวัง ซึ่งใช้เวลามากกว่าขั้นตอนอื่น รวมทั้งท่านจะต้องทราบเป็นอย่างดีว่าแต่ละเรคอร์ดนั้นสมควรที่จะไปลิงค์กับกลุ่ม Entity ใด นอกจากนี้แล้วแต่ละ Entity ยังสามารถที่จะลิงค์กับเรคอร์ดได้อย่างไม่จำกัดจำนวน ยกเว้นเรคอร์ดที่ซ้ำกันจะกระทำไม่ได้

หลักของการลิงค์

หลักการง่าย ๆ ของการลิงค์ก็คือท่านจะต้องไปค้นหาเรคอร์ดที่ท่านต้องการจะลิงค์ แล้วตั้งค่าให้เป็น Current Record ก่อน (Set Row...) จากนั้นจึงไปทำการ Make link ให้ Pick เข้ากับ Entity ที่ต้องการนั้น และทำเช่นนี้ไปเรื่อย ๆ จนครบตามจุดประสงค์... เพื่อความสะดวกต่อความเข้าใจ ผมจะขอแบ่งขั้นตอนการลิงค์ออกเป็นขั้นตอนดังต่อไปนี้

ขั้นตอนที่ 1

เปิด Drawing file ที่ชื่อว่า "ASETUT.DWG" ขึ้นมาอีกครั้ง ดังแสดงในรูปที่ 4

ขั้นตอนที่ 2

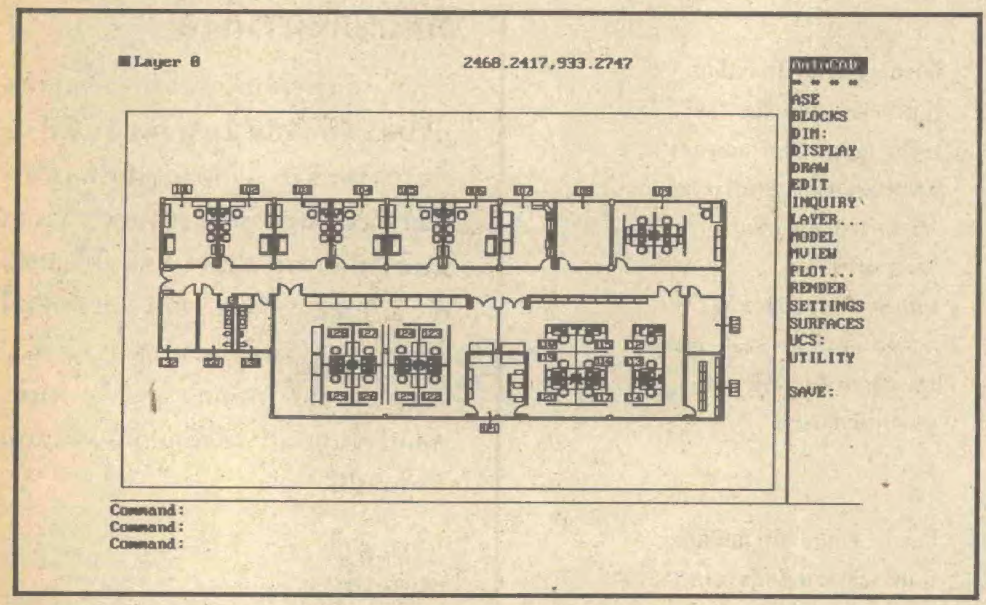
Initialize ASE on Pull-Down Menu ให้ป้อนข้อมูลเมื่อโปรแกรมถาม Username และ Password ในที่นี้ให้กด Enter ผ่านไปเลย เพราะยังไม่มีข้อมูลในส่วนนี้

ขั้นตอนที่ 3

ใช้คำสั่ง ASESETTABLE เรียกไดอะล็อกบ็อกซ์ของการเลือก Table ในดาต้าเบสที่สนใจให้เป็น Table ปัจจุบัน (Current Table) ซึ่งจะมีสภาพพร้อมที่จะใช้งาน

ขั้นตอนที่ 4

เลือก "EMPLOYEE" เป็น Table ปัจจุบัน โดยการ Pick ด้วยอุปกรณ์ชี้ในรายการที่ให้เลือกจะปรากฏแถบสว่างขึ้นมา พร้อม Pick ที่ OK เพื่อยืนยันการเลือกพร้อมกับ Exit เพื่อออกจากไดอะล็อกบ็อกซ์



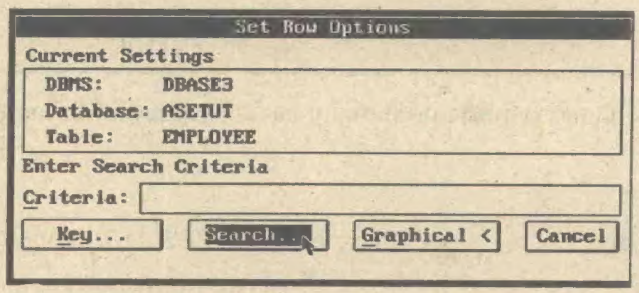
รูปที่ 4 Drawing ตัวอย่างที่ชื่อว่า ASETUT.DWG

ขั้นตอนที่ 5

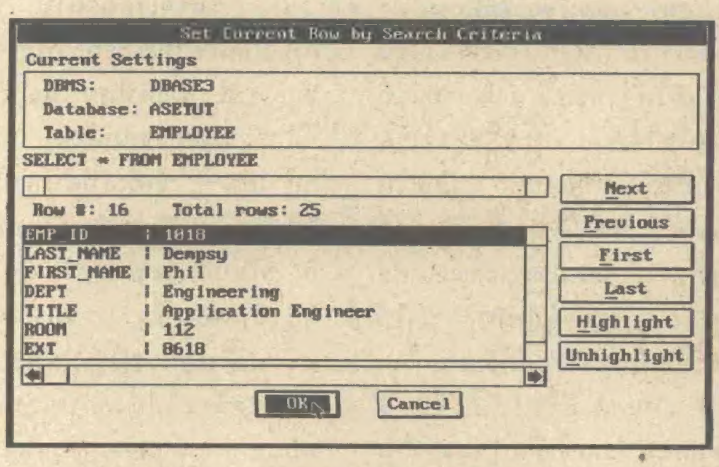
ใช้คำสั่ง ASESETROW เรียกไดอะล็อกบ็อกซ์เพื่อเซตเรคอร์ดปัจจุบันตามที่ต้องการเสียก่อน ดังแสดงในรูปที่ 5 ซึ่งในไดอะล็อกบ็อกซ์ของเราจะประกอบไปด้วย 3 ตัวเลือก (buttons) อันประกอบด้วย key, search..., Graphical< ให้ท่านเลือก search... เพื่อแสดงเรคอร์ดหรือ Row ทั้งหมดที่มีอยู่ใน Table ปัจจุบัน (EMPLOYEE.DBF)

ขั้นตอนที่ 6

โปรแกรมจะแสดงซับไดอะล็อกบ็อกซ์ซ้อนทับดังแสดงในรูปที่ 6 และใช้ปุ่ม Next เพื่อแสดงเรคอร์ดถัดไปซึ่งจะเริ่มตั้งแต่ Row#1 (Record) ไปเรื่อย ๆ จนถึง Row สุดท้าย ให้ค้นหา Row#16 ที่เป็นรายชื่อพนักงานหมายเลข 1018 กำหนดให้เป็นเรคอร์ดปัจจุบัน ก็ให้เลือก OK ก็ถือว่าพร้อมที่จะลิงค์กับ Entity ได้



รูปที่ 5 ไดอะล็อกบ็อกซ์ของการเลือกเรคอร์ดปัจจุบัน



รูปที่ 6 ไดอะล็อกบ็อกซ์ของการตั้งค่าเรคอร์ดปัจจุบัน

ขั้นตอนที่ 7

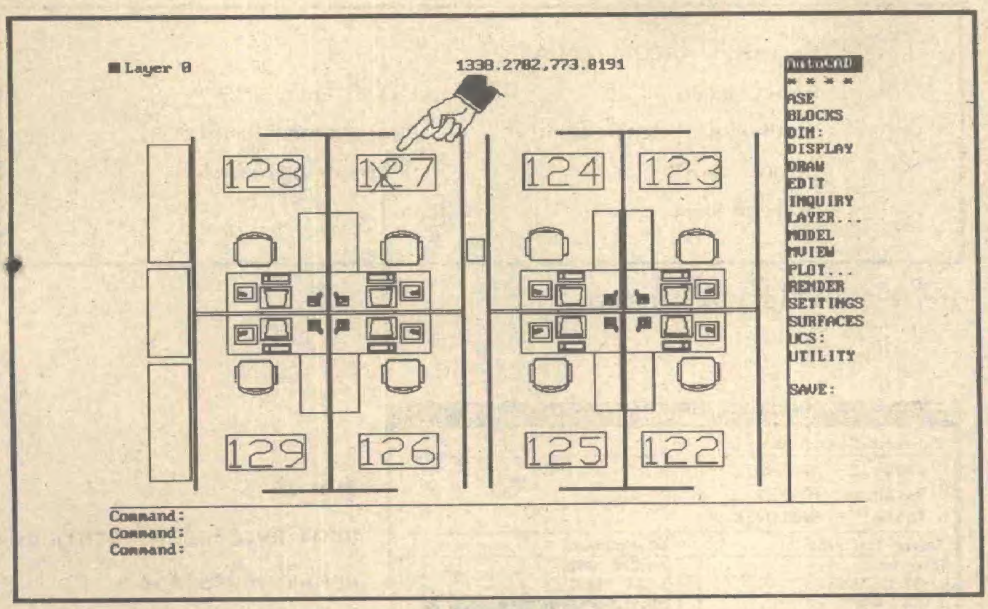
สร้างการลิงค์ระหว่าง Graphic Entity กับเรคอร์ดปัจจุบันด้วยคำสั่ง

```
Command : ASEMAKELINK
```

ซึ่งโปรแกรมจะถามให้เราเลือก Pick บนจอภาพ

```
Select object:          <Pick>
```

ให้ Pick ที่ Text ที่เป็นตัวเลขห้องเบอร์ "127" (ในรูปที่ 7) แล้วกด Enter และในขณะนี้ Entity ที่เป็นหมายเลขห้อง "127" ก็ได้สร้างลิงค์ระหว่างเรคอร์ดกับ Entity ไว้เรียบร้อยแล้ว



รูปที่ 7 แสดงการ link Entity กับเรคอร์ดปัจจุบัน

ขั้นตอนที่ 8

เรคอร์ดในปัจจุบันยังสามารถลิงค์กับ Entity อื่น ๆ ได้อย่างไม่จำกัดจำนวน และในแต่ละ Entity ยังสามารถสร้างการลิงค์ได้มากกว่า 1 เรคอร์ดโดยไม่จำกัดว่าจะลิงค์กับ Table ของฐานข้อมูลใด

ขั้นตอนที่ 9

ท่านจะสามารถตรวจสอบการลิงค์ว่าเป็นไปอย่างถูกต้องหรือไม่โดยใช้คำสั่ง

```
Command : ASEQVIEW
```

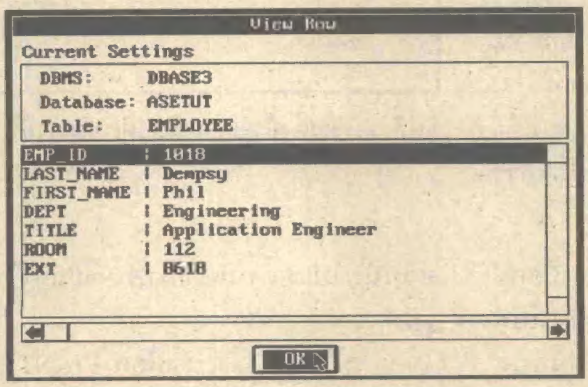
แล้วโปรแกรมจะให้เรา "Select object:" ก็ให้ Pick เลือก Entity ตรงเท็กซ์ "127" อีกครั้ง ก็จะแสดงข้อมูลทั้งหมดของการลิงค์ทั้งหมดบนไดอะล็อกบ็อกซ์ให้ท่านได้ทราบว่า Entity นั้นลิงค์อยู่ทั้งหมดกี่เรคอร์ด แล้วให้ใช้ปุ่ม Next ไปยังเรคอร์ดที่ 2 (link#2:) และ Pick OK ก็จะเกิดไดอะล็อกของวิวลิงค์ดังแสดงในรูปที่ 8 ก็เป็นอันว่า Entity ของท่านได้ลิงค์กับเรคอร์ดหรือ Row ถูกต้องเรียบร้อยแล้ว ซึ่งท่านจะยังสามารถให้เรคอร์ดตัวอื่น ๆ เข้ามาลิงค์กับ Entity นี้ได้อีก ก็จะกลายเป็น link#3: Total link:3 เช่นนี้ไปเรื่อย ๆ

ลดขั้นตอน

นอกจากที่ได้กล่าวมาแล้วในตัวเลือกย่อยของ ASE ที่ *"Quick Link..."* จะสามารถใช้แทนทั้งสองขั้นตอนได้เลย แทนที่จะต้องเซตเรคอร์ดกันก่อนแล้วจึงทำการลิงค์กันอีกทีภายหลัง โดยตัวเลือก Quick link...นี้ จะให้ท่านเลือกเรคอร์ดจากไดอะล็อกบ็อกซ์ก่อนจนเป็นที่พอใจแล้วถึงจะเลือกลิงค์กับ object: ไปเลย ไม่ต้องมาสะดุดให้เสียจังหวะ ผมว่าน่าจะมีประสิทธิภาพมากกว่า

ติดป้ายแอตทริบิวต์

ก็ข้ามกับเข้ามาสู่ขั้นตอนที่ 5 แล้วนะครับ ขั้นตอนนี้เป็นการกล่าวถึงการติดป้ายแอตทริบิวต์ที่บางท่านอาจจะเรียกว่าป้ายผนึก แต่ผมขอเรียกทับศัพท์ดีกว่า (จะได้ไม่สับสน) ปกติในการแสดงรูปกราฟิกใน Drawing ของท่านบางครั้งก็ต้องมีความจำเป็นที่จะแสดงรายละเอียดหรือสเปคบางอย่างกำกับ Entity ไว้ด้วย เพื่อจะได้เกิดความสมบูรณ์หรือสะดวกในการศึกษาแบบ Drawing นั้น ๆ ASE จึงเปิดโอกาสให้ท่านสามารถนำข้อมูลในดาต้าเบส (ที่ลิงค์) มาแสดงในรูปแบบแอตทริบิวต์ และแอตทริบิวต์ที่พูดถึงนี้ มันจะคล้ายตราประทับให้รู้ว่าชิ้นใดเป็นของใครบ้าง คล้ายกับเราต้องใส่รหัสไปรษณีย์นั่นแหละครับ จะได้แยกกองแยกพวกได้ถูก



รูปที่ 8 ไดอะล็อกบ็อกซ์ของกาววิว เข้าไปดูเรคอร์ดของ Entity

แต่ต้องเข้าใจด้วยว่าป้ายแอตทริบิวต์นี้จะไม่ปรากฏให้เห็น จะต้องไปสั่งให้มันแสดงตัวของมันออกมาก่อน ดังนั้นดาต้าเบสที่ลิงค์เข้าไปกับ Entity ก็เปรียบได้กับการนำป้ายแอตทริบิวต์ต่าง ๆ ไปแปะไว้ และถ้าเมื่อใดต้องการให้มันแสดงตัวขึ้นมาใน Drawing ก็ต้องเข้าไปสั่งให้มันปรากฏตัว... ลองทำดูสิครับ จะได้รู้ว่าเป็นจริงอย่างที่กล่าวไว้หรือเปล่า

ขั้นตอนการติดป้ายแอตทริบิวต์

เริ่มด้วยการพิมพ์คำสั่งเพื่อสร้างดาต้า

```
Command: ASEQMAKEDA
```

คำสั่งนี้จะเป็นการดึงข้อมูลแอตทริบิวต์ที่เราต้องการมาจากเรคอร์ดที่เราลิงค์กับ Entity นั้น ๆ ไปแล้วให้ออกมาเป็นในรูปของเท็กซ์ตรงตำแหน่งที่ต้องการ และเมื่อใช้คำสั่งนี้โปรแกรมจะถามถึง Entity ที่ต้องการจะสร้างดาต้า

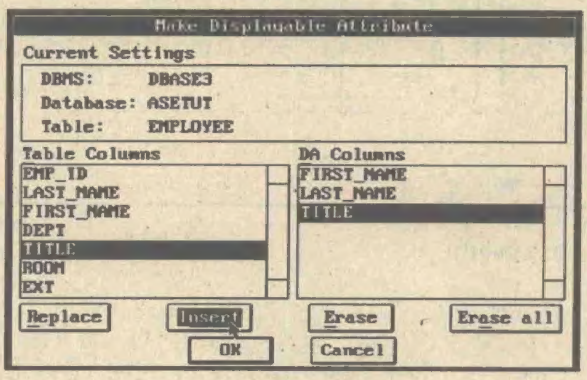
```
Select object : <Pick ที่เท็กซ์ข้อความ
                "127">
```

ให้ Pick เลือก Entity ที่เท็กซ์ "127" ก็จะเกิดไดอะล็อกบ็อกซ์ขึ้นมา จากนั้นก็ให้ใช้ตัวเลือกเน็ตไปเป็น *link #2* จึงจะ OK (อ้างอิงในเรคอร์ดที่ 2)

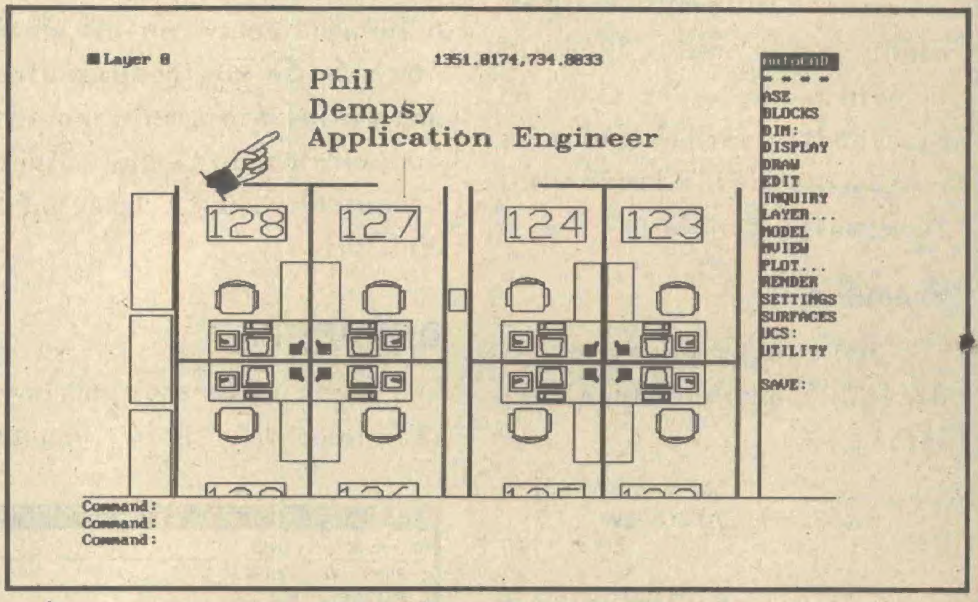
หลังจากนั้นโปรแกรมก็จะถามถึงตำแหน่ง ความสูง รวมทั้งมุมที่ต้องการหมุนเท็กซ์ด้วย ดังแสดงในรูปที่ 9

```
Command: ASEQMAKEDA
Select object:                          ; Pick ที่ Entity "127"
Justify/Style/<Start point>:            ; Pick ตำแหน่งที่วางเท็กซ์
Height <0.28>:                          ; ป้อนค่าความสูงที่ต้องการ
Rotation angle <0>:                     Enter
```

รูปที่ 9 คำถามที่โปรแกรมจะต้องถาม



รูปที่ 10 แสดงไดอะล็อกบ็อกซ์และการแสดงแอตทริบิวต์ด้วย ASE



รูปที่ 11 แสดงป้ายแอตทริบิวต์ภายหลังที่ให้แสดงข้อมูลออกมา

หลังจากนี้ก็จะเกิดไดอะล็อกบ็อกซ์ขึ้นมาโต้ตอบ ให้ท่านเลือกคอลัมน์บนเรคอร์ดที่ต้องการ เพื่อที่จะนำข้อมูลส่วนนั้นไปแสดงบนจอภาพ ดังแสดงในรูปที่ 10

ให้ใช้ปุ่ม Insert เลือก *FIRST_NAME, LAST_NAME* และ *TITLE* เพื่อนำข้อมูลของเรคอร์ดในส่วนนั้นไปปรากฏบนจอภาพดังแสดงในรูปที่ 11

แต่ถึงอย่างไรก็ตาม ถ้าเกิดดาต้าเบสของท่านที่เป็นเรคอร์ดมีการแก้ไขไปแล้ว ถามว่าป้ายแอตทริบิวต์ที่เราสั่งให้แสดงออกมานั้นจะมีการแก้ไขตามกันไปด้วยหรือไม่ ก็ขอตอบได้เลยครับว่า *"ไม่ได้"* แต่จะต้องใช้คำสั่ง

```
Command: ASERELOADDA
```

เพื่อทำการอัพเดตป้ายแอตทริบิวต์ไปเป็นไปตามเรคอร์ดล่าสุดตามกันไปด้วยนะครับ

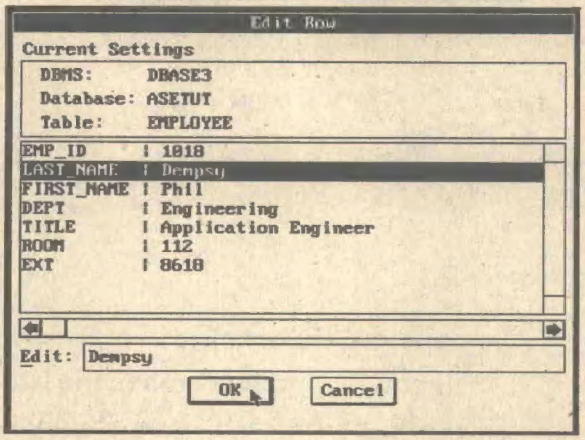
ปรับแก้ดาต้าเบสด้วยออโต้แคด

ขั้นตอนนี้เป็นขั้นตอนที่ 6 ที่กล่าวถึงการแก้ไขดาต้าเบสด้วยออโต้แคดโดยตรง ซึ่งจะทำให้สะดวกมากขึ้น เหมาะสำหรับการแก้ไขเล็ก ๆ น้อย ๆ เฉพาะจุดแทนที่จะต้องไปพึ่ง DBMS ภายนอกโปรแกรม ทำงานก็ทำตรงนั้นไปเลย โดยใช้คำสั่ง

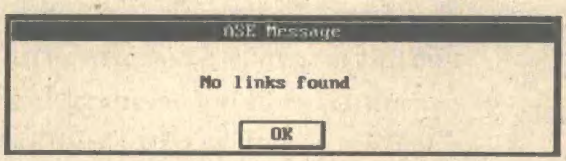
```
Command: ASEWQEDIT
```

โปรแกรมจะให้ท่านเลือก Entity *"Select Object"* โดยการ Pick Entity ที่ต้องการแก้ไขเรคอร์ด สมมติ Pick ตรงที่ Entity "127" แล้วเลือก Link#2 พร้อม OK ก็จะเกิดซับไดอะล็อกบ็อกซ์ขึ้นมาให้เราแก้ไข (ดังแสดงในรูปที่ 12) ซึ่งจะกระทำได้โดยง่าย เพียงใช้อุปกรณ์ชี้ Pick ลงในช่อง Edit ก็จะสามารถแก้ไขข้อมูลตามที่ต้องการได้เลย

แต่ถ้าสมมติว่าท่านไปเลือก Entity ที่ยังไม่ได้ลิงค์กับดาต้าเบสใด ๆ เลย ก็จะเกิดข้อผิดพลาดขึ้น (error) และแจ้งให้ทราบดังแสดงในรูปที่ 13



รูปที่ 12
ไดอะล็อกบ็อกซ์ของการแก้ไขเรคอร์ดในแต่ละ Entity ที่เลือก



รูปที่ 13
ข้อความแสดงความผิดพลาดเมื่อเลือก Entity ที่ยังไม่มีการลิงค์

เติมความสมบูรณ์ด้วย SQL

SQL เป็นลักษณะหนึ่งที่ผมขอยอมรับว่า *"วิเศษ"* ที่ผ่าน ๆ มาจะเป็นการกล่าวถึงเฉพาะการเข้าไปดู แก้ไข, ลิงค์ดาต้าเบสเท่านั้น ยังไม่ใช้เป็นการสอบถาม ค้นหาข้อมูล หรือการวิเคราะห์แต่อย่างใด SQL นี้ออโต้แคดได้นำเอาภาษา SQL มาผนวกกับความสามารถทางด้านกราฟิก เพื่อการวิเคราะห์ข้อมูลให้ได้ตามความต้องการ

SQL ความจริงเป็นภาษาหนึ่งที่ทำหน้าที่ในการค้นหาและสอบถามข้อมูล เป็นภาษาที่ค่อนข้างง่ายเมื่อเปรียบเทียบกับภาษาโครงสร้างตัวอื่น ๆ พร้อมกับเป็นที่นิยมแพร่หลายโดยเฉพาะทางด้านดาต้าเบส SQL นี้จะทำให้ท่านสามารถติดต่อกับ Table ของดาต้าเบสได้โดยตรง โดยไม่ต้องออกจากออโต้แคดและในหัวข้อนี้จะเป็นการกล่าวถึงคำสั่งพื้นฐาน (Query) ที่ใช้กันบ่อย ๆ นั่นก็คือ คำสั่ง SELECT และคำสั่ง INSERT INTO ที่ใช้สำหรับสอบถามข้อมูลและเพิ่มข้อมูลลงสู่ Table

การค้นหาบน SQL ด้วยคำสั่ง SELECT

ท่านสามารถใช้คำสั่ง SELECT ในการค้นหาและสอบถามข้อมูลจาก Table ข้อมูล โดยจะเป็นการค้นหาข้อมูลตามจุดประสงค์ป้อนเข้าไป คำสั่งนี้จะต้องใช้คู่กับพารามิเตอร์ "FROM" เสมอ ในลักษณะเลือก *"อะไร?"* จาก Table คำว่า "อะไร?" นี้จะเป็นสิ่งที่ท่านต้องการจะค้นหาตามรูปแบบดังแสดงในรูปที่ 14

- **Select Clause** : เป็นคำสั่ง SQL สำหรับสอบถามข้อมูล โดยให้เลือกเอาเฉพาะข้อมูลคอลัมน์ ซึ่งอาจจะมีหลาย ๆ คอลัมน์ก็ได้ หรือบางครั้งก็ใช้เครื่องหมาย (*) ก็หมายความว่าเป็นการเลือกคอลัมน์ในทุก ๆ คอลัมน์

```
SELECT      ชื่อคอลัมน์                ; เลือกสิ่งที่ต้องการ
FROM        ชื่อตามข้อมูล Table        ; จาก Table อะไร
```

รูปที่ 14 รูปแบบที่จะต้องค้นหาจาก Table

ที่มีอยู่ใน Table นั้นเอง เช่น

```
Select * from EMPLOYEE...
```

● **From Clause** : เป็นประโยคที่สองของคำสั่ง Select เพื่อเป็นการกำหนดว่าจะให้เลือกข้อมูลที่สอบถามจาก "Select Clause" ได้จาก Table ใดบ้างที่ต้องการจะสอบถามข้อมูล และจะต้องตามด้วยการกำหนดคอลัมน์และชื่อ Table ในคำสั่ง Query อย่างชัดเจนและแน่ชัดด้วย ยกตัวอย่างเช่น

```
– Select * from INVENTRY
Where Price>500
– Select * from EMPLOYEE
where FIRST_NAME='VITTAYA'
– Select * from COMPUTER
where HardDisk='120 Mb'
```

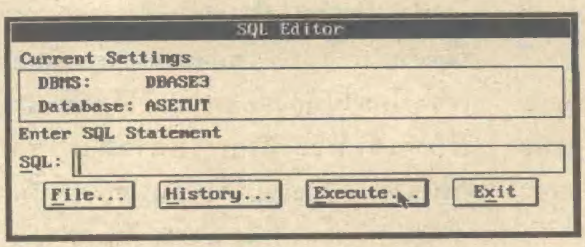
การใช้งาน SQL บนคำสั่ง SELECT

สมมติว่าให้เราทำการลิงค์เรคอร์ดทั้งหมดเข้ากับ Entity ต่าง ๆ จนครบถ้วนตามจุดประสงค์เรียบร้อยแล้ว ซึ่งท่านจะมองไม่เห็นข้อมูลทั้งหมดที่ซ่อนอยู่ภายใน Entity เลย (รูปตัวอย่าง Drawing ที่ชื่อ ASETUT. DWG) รูปบนหน้าจอของท่านจะเป็นรูปแปลนของสำนักงานแห่งหนึ่งที่ประกอบด้วยตัวอาคาร ห้องทำงานที่มีหมายเลขกำกับ เฟอร์นิเจอร์ คอมพิวเตอร์ และอื่น ๆ ยกตัวอย่างในกรณีเฟอร์นิเจอร์ที่มีอยู่มากมาย ทั้งสเปคหรือข้อมูลจำเพาะก็คละเคล้ากันไป ถ้าเกิดท่านต้องการจะทราบว่าเฟอร์นิเจอร์ทั้งหมดนี้มีกี่ชิ้นที่ราคาสูงกว่า 5,000 บาทขึ้นไป อยู่ตำแหน่งไหนบ้างของสำนักงาน ใครเป็นผู้ใช้งานอยู่ และบริษัทใดที่ผลิตชิ้นเหล่านี้ขึ้นมา หรือจากพนักงานที่มีจำนวนเกือบ 40 คน จะมีสักกี่คนที่ทำงานอยู่ในตำแหน่งวิศวกร ทำงานอยู่ห้องเบอร์อะไร อยู่ตรงจุดไหนของสำนักงาน มีชื่ออะไรบ้าง และเบอร์โทรศัพท์ที่ติดต่อได้เบอร์อะไร ให้แสดงข้อมูลที่ต้องการบนจอภาพ

เท่านี้ละครับก็คงสร้างความปวดหัวให้แก่ท่านไม่น้อยเลยทีเดียว ซึ่งถ้าเป็นสำนักงานใหญ่ ๆ แล้วคงไม่ต้องพูดถึงกันเลย แต่ SQL จะช่วยให้ท่านสามารถสอบถามและค้นหาข้อมูลได้อย่างสะดวกและรวดเร็ว ด้วยการใช้คำสั่ง

```
Command: ASESQLED
```

โปรแกรมจะเปิดไดอะล็อกบ็อกซ์ขึ้นมาตอบโต้กับท่าน ดังแสดงในรูปที่ 15



รูปที่ 15
ไดอะล็อกบ็อกซ์ของ SQL ที่รอรับการป้อนข้อมูล

ถ้าท่านสังเกตไดอะล็อกบ็อกซ์ในรูปที่ 15 จะมีการแสดง DBMS และดาตาเบสที่ใช้งานอยู่ให้ท่านทราบพร้อมกับเกิดพรอมต์ขึ้นรอรับคำสั่ง SQL โดยจะมีปุ่มช่วยคือ History... เอาไว้คอยบันทึกเรื่องราวที่เคยป้อนเข้าไป Execute... จะเป็นการนำคำสั่ง SQL ที่ท่านได้ป้อนเข้าไปทำการคำนวณและแสดงผลออกมา

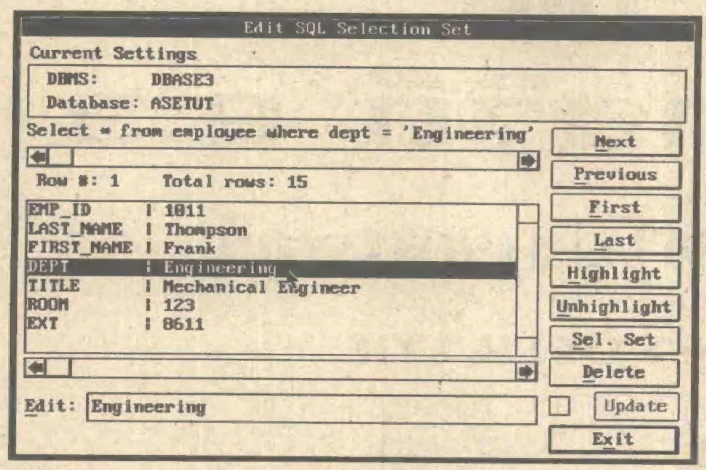
ทดสอบประสิทธิภาพ ASE

ก็คงถึงตอนที่จะได้ทดสอบประสิทธิภาพของ ASE ที่ซ่อนไว้เสียที เช่น ถ้าต้องการข้อมูลเรื่องพนักงานที่ทำงานตำแหน่งวิศวกร พร้อมทั้งแสดงตำแหน่งบนจอภาพด้วย ก็ให้ป้อนข้อความด้วยการพิมพ์ลงไปดังนี้

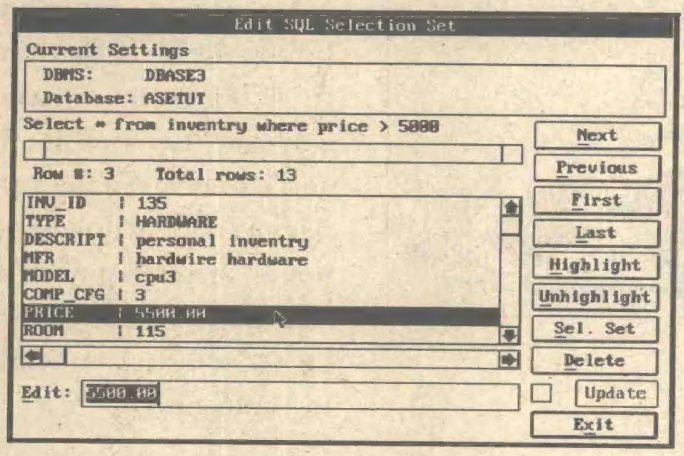
```
Select * from employee where dept
= 'Engineering'
```

แล้ว Pick ที่ปุ่ม Execute เพื่อทำงาน (ระวังเรื่องตัวพิมพ์เล็กพิมพ์ใหญ่ด้วยครับ) ก็จะได้ผลลัพธ์ดังแสดงในรูปที่ 16

ตอนนี้ท่านก็ทราบข้อมูลที่ต้องการแล้ว พนักงานที่เป็นวิศวกรมีอยู่ด้วยกันทั้งหมด 15 คน มีชื่อว่า... อยู่ในห้อง... เบอร์โทรศัพท์... ฯลฯ ให้ใช้ปุ่ม Next ไปยังข้อมูลของคนที่สองและที่สามไปเรื่อย ๆ และถ้าต้องการทราบว่าอยู่ตรงตำแหน่งไหนของสำนักงานบน



รูปที่ 16 แสดงข้อมูลทั้งหมดที่ค้นหาด้วย SQL Command



รูปที่ 17 ไดอะล็อกบ็อกซ์ของการสอบถามด้วย SQL อีกเช่นกัน

หน้าจอ ก็ใช้ปุ่ม Hightlight ก็จะทำให้ Entity ที่มีเรคอร์ดส่วนนั้นอยู่ปรากฏเป็นเส้นสว่าง (เส้นประ) บนจอภาพ ซึ่งถ้ามีขนาดเล็กท่านอาจจะสังเกตเห็นได้ไม่ชัดเจนนัก แต่รับรองว่าเกิดขึ้นแน่ครับ

หรือจะลองอีกสักตัวอย่างหนึ่งก็ได้ คงจะทำให้ชัดเจนขึ้น ตอนนี้ Exit กลับมายัง SQL Editor เดิมก่อน คราวนี้ต้องการทราบว่า เฟอร์นิเจอร์ในสำนักงานแห่งนี้มีกี่ชิ้นที่ราคาสูงกว่า 5,000 บาทขึ้นไป และอยู่ตำแหน่งไหนบ้างก็ให้พิมพ์คำสั่ง SQL

```
Select * from inventry
where price>5000
```

ก็จะได้ข้อมูลทั้งหมดที่ถูกสอบถามปรากฏบนไดอะล็อกบ็อกซ์ ทำให้ทราบว่าทั้งสำนักงานแห่งนี้ มีอยู่ด้วยกันทั้งหมด 13 ชิ้นเท่านั้นที่ราคาเกิน 5,000 ดังแสดงในรูปที่ 17

และถ้าต้องการทราบตำแหน่งของชิ้นเฟอร์นิเจอร์นี้ว่าอยู่ตำแหน่งไหน ก็ใช้ตัวเลือก Hightlight... อีกเช่นกัน

ยังมีอีก

ความจริงเรื่องราวของ SQL นี้ยังมีอีกมาก และเป็นเรื่องราวเรื่องใหญ่เรื่องหนึ่งที่ยังมีคำสั่งอีกมากมายที่ช่วยให้การสอบถามหรือค้นหาข้อมูลที่เต็มไปด้วยประสิทธิภาพ ไม่ว่าจะเป็นเรื่องราวที่ซับซ้อนเพียงใดก็ตาม และที่น่าสนใจมากก็คือ ท่านสามารถที่จะเขียนภาษา SQL ในลักษณะเป็นแบบโปรแกรมที่คล้ายการสั่งการด้วย AutoLISP โดยการจัดเก็บโปรแกรมลงในไฟล์ แล้วใช้ปุ่ม File... เพื่อเรียกขึ้นมา Execute ในแบบ Interpreter (ที่เป็นการคอมพายล์แบบทีละบรรทัด) ได้ ซึ่งจะทำให้การทำงานของ ASE มีประสิทธิภาพอย่างเต็มที่เลยทีเดียว

ส่งท้าย

ยังครับยังไม่จบลงแค่นี้ ในคราวหน้าคงจะเป็นเรื่องราวของคำสั่งบางคำสั่งของ SQL ที่น่าสนใจอีก ความจริงผมอยากจะอธิบายให้ละเอียดทุกคำสั่งมากกว่านี้ แต่ก็ด้วยข้อจำกัดบางประการ ไม่ว่าจะเป็นเรื่องของจำนวนหน้าหนังสือ ระยะเวลา ไปจนถึง Schedule ที่ได้วางเอาไว้ ทำให้ต้องตัดใจแบ่งเป็นตอน ๆ และก็คงจะจบบริบูรณ์แน่ในฉบับหน้า และถ้าหากต้องการข้อมูลเพิ่มเติมหรือติดขัดตรงจุดไหน ก็ขอให้ติดต่อกับผมได้โดยตรงนะครับ ฮอตไลน์ 4881068 สำหรับฉบับนี้ *สวัสดีครับ*